

CLAIMS

What is claimed is:

- 1 1. A computer-implemented method for instrumentation of selected functions in an
2 executable program, the selected functions initially occupying an original address space of
3 the executable program, comprising:
4 generating instrumented versions of selected functions in relocation address space
5 during program execution;
6 when a function is called by an instrumented version of a selected function within
7 the relocation address space resulting in a first return-pointer value in the relocation
8 address space, identifying a location in the original address space corresponding to the
9 first return-pointer value as an original return-pointer value, associating the first return-
10 pointer value with the original return-pointer value, substituting references to the original
11 return-pointer value for references to the first return-pointer value, and replacing an
12 instruction at the address indicated by the original return-pointer value with a breakpoint;
13 and
14 when the breakpoint is encountered upon return of control at the original return-
15 pointer value, obtaining the first return-pointer value associated with the original return-
16 pointer value, and transferring control to an instruction at the address referenced by the
17 first return-pointer value.
- 1 2. The method of claim 1, further comprising identifying RP-sensitive functions as
2 the selected functions, wherein RP-sensitive functions are those functions that require a
3 return pointer value in the original address space of the executable program.

1 3. The method of claim 2, further comprising:
 2 patching entry points of the RP-sensitive functions with RP-entry breakpoints; and
 3 generating an instrumented version of an RP-sensitive function upon encountering
 4 the RP-entry breakpoint at the entry point of the RP-sensitive function.

1 4. The method of claim 3, wherein each instrumented version of an RP-sensitive
 2 function has a corresponding original version function in the original address space,
 3 further comprising replacing instructions at entry points of the instrumented versions of
 4 the RP-sensitive functions with branch instructions targeting corresponding instrumented
 5 versions of the RP-sensitive functions.

1 5. The method of claim 4, further comprising identifying the RP-sensitive functions
 2 through analysis of code segments within the executable program.

1 6. The method of claim 4, further comprising identifying the RP-sensitive functions
 2 through an input list of identifier codes associated with RP-sensitive functions.

1 7. The method of claim 2, wherein each instrumented version of an RP-sensitive
 2 function has a corresponding original version function in the original address space,
 3 further comprising replacing instructions at entry points of the instrumented versions of
 4 the RP-sensitive functions with branch instructions targeting corresponding instrumented
 5 versions of the RP-sensitive functions.

1 8. The method of claim 2, further comprising identifying the RP-sensitive functions
 2 through analysis of code segments within the executable program.

1 9. The method of claim 2, further comprising identifying the RP-sensitive functions
2 through an input list of identifier codes associated with RP-sensitive functions.

1 10. The method of claim 2, further comprising:
2 generating the relocation address space;
3 inserting RP-entry breakpoints at entry points of the RP-sensitive functions; and
4 upon encountering an RP-entry breakpoint during execution of the executable
5 program, generating an instrumented version of the RP-sensitive function associated with
6 the RP-entry breakpoint, and replacing the RP-entry breakpoint with a branch instruction
7 that targets the instrumented version of the RP-sensitive function.

1 11. A computer-implemented method for instrumentation of selected functions in an
2 executable program, the selected functions initially occupying an original address space of
3 the executable program, comprising:
4 generating relocation address space;
5 identifying RP-sensitive functions in the executable program, wherein RP-sensitive
6 functions are those functions that require a return pointer value in the original address
7 space;
8 inserting RP-entry breakpoints at entry points of the RP-sensitive functions;
9 upon encountering an RP-entry breakpoint during execution of the executable
10 program, generating an instrumented version of the RP-sensitive function associated with
11 the RP-entry breakpoint, and replacing the entry point of the RP-sensitive function in the
12 original address space with a branch instruction that targets the instrumented version of the
13 RP-sensitive function;

14 when an instrumented version of RP-sensitive function is called from a function in
15 the relocation address space whereby a return-pointer register stores a first return-pointer
16 value within the relocation address space, identifying a location in the original address
17 space corresponding to the first return-pointer value as an original return-pointer value,
18 associating the first return-pointer value with the original return-pointer value, storing the
19 original return-pointer value in the return-pointer register, and replacing an instruction at
20 the address indicated by the original return-pointer value with an RP-return breakpoint;
21 and

22 when the RP-return breakpoint is encountered upon return of control at the original
23 return-pointer value, obtaining the first return-pointer value associated with the original
24 return-pointer value, restoring the first return-pointer value to the return-pointer register,
25 and transferring control via the return pointer register.

1 12. The method of claim 11, further comprising identifying the RP-sensitive functions
2 through analysis of code segments within the executable program.

1 13. The method of claim 11, further comprising identifying the RP-sensitive functions
2 through an input list of identifier codes associated with RP-sensitive functions.

1 14. An apparatus for instrumentation of selected functions in an executable program,
2 the selected functions initially occupying an original address space of the executable
3 program, comprising:

4 means for generating instrumented versions of selected functions in relocation
5 address space during program execution;

10012808-1

6 means, responsive to a call to an instrumented version of a selected function from
7 within the relocation address space whereby a first return-pointer value is within the
8 relocation address space, for identifying a location in the original address space
9 corresponding to the first return-pointer value as an original return-pointer value,
10 associating the first return-pointer value with the original return-pointer value, substituting
11 references to the original return-pointer value for references to the first return-pointer
12 value, and replacing an instruction at the address indicated by the original return-pointer
13 value with a breakpoint; and
14 means, responsive to encountering the breakpoint upon return of control at the
15 original return-pointer value, for obtaining the first return-pointer value associated with
16 the original return-pointer value, and transferring control to an instruction at the address
17 referenced by the first return-pointer value.